常微分方程式の初期値問題の精度保証付き数値解法における技術的諸問題(数値解析とそのアルゴリズム)

AUTHOR(S):

雨宮, 治郎

CITATION:

雨宮, 治郎. 常微分方程式の初期値問題の精度保証付き数値解法における技術的諸問題(数値解析とそのアルゴリズム). 数理解析研究所講究録 1992, 791: 126-139

ISSUE DATE:

1992-06

URL:

http://hdl.handle.net/2433/82687

数理解析研究所講究録
第 791 巻 1992 年 126-139

常微分方程式の初期値問題の精度保証付き数値解法における技術的諸問題

東京大学工学部計数工学科　　雨宮治郎　(Jiro AMEMIYA)

1　はじめに

自動微分の手法 [1],[2] を用いると，常微分方程式の初期値問題のべき級数解をつくり，それに保証区間を与えることが容易に自動的にできる [3],[4],[5],[6]．その際未知関数の値の変動区間についての何らかの見積もりが必要になる．また，効率の良い次数及びステップ幅の自動変更のルールや，効率良く狭い保証区間を与える方法などの実験結果の一部についても既に発表したが [7],[8]，本稿ではこれらの技術的問題についてのその後の検討結果を示す．丸め誤差の影響も同様の方法で捉えることができるが，ここでは一応無視している．

2　Taylor 展開法

正規形の常微分方程式の初期値問題 (あるいは何らかのアルゴルズムにより正規形に直るような問題)

$$y' = f(y), \quad y(t_0) = y_0 \tag{1}$$

が与えられたとする (以下自律系に限る)．自動微分の手法を用いると，$t = t_0$ における y の t に関する Taylor 展開の係数が初期値 y_0 と f を計算する手続きとから計算できる．このとき，Taylor 展開の係数は低い次数の係数から順に得られ，p 次の係数を計算するのに必要な手間は $O(p^2)$ である．以後，y の p 次の Taylor 展開係数を，引数 y_0 から計算したことを明示するために $y^{<p>}(y_0)$ と記す．

さて，ステップ点 $t = t_n$ における未知関数の値 $y_n = y(t_n)$ が与えられたとき，次のステップ点 $t = t_{n+1} = t_n + h$ における未知関数の値 $y_{n+1} = y(t_{n+1})$ は Φ と $[z_{n+1}]$ を

$$\begin{cases} \Phi(\xi; h) & := & y^{<1>}(\xi) + hy^{<2>}(\xi) + \cdots + h^{p-1}y^{<p>}(\xi), \\ [z_{n+1}] & := & h^{p+1}y^{<p+1>}(Y_n) \end{cases} \tag{2}$$

と定義すれば次を満たす:

$$y_{n+1} \in y_n + h\Phi(y_n; h) + [z_{n+1}]. \tag{3}$$

自動微分法を応用すれば $\Phi(\xi; h)$ を計算する手続きが構成できる．Y_n は y の $[t_n, t_n + h]$ における変動区間の見積もりであり，剰余項は $[z_{n+1}]$ に含まれる．

(3) を見ると，あるステップ点での数値解が点で与えられても，次のステップ点においては解を含む保証区間しか得られないことがわかる．よって本稿では，$t = t_n$ における真の解を含む保証区間 $[y_n]$ から，$t = t_{n+1}$ における真の解を含む保証区間 $[y_{n+1}]$ を計算する Taylor 展開法について検討する．

$t = t_n$ における保証区間 $[y_n]$ から，$t = t_{n+1}$ における保証区間 $[y_{n+1}]$ を計算する方法で最も簡単なのは次の方法である:

$$[y_{n+1}] := [y_n] + h\Phi([y_n]; h) + [z_{n+1}]. \tag{4}$$

しかし，公式 (4) を用いると $[y_{n+1}]$ の区間幅は $[y_n]$ の区間幅より常に大きくなり保証区間の幅が増大する一方である．

保証区間が増大する一方であるという事実は公式 (4) に依存する．これに対し改善が期待できる方法は，保証区間 $[y_n]$ を中点 $\hat{y}_n$ と全離散化誤差の評価区間 $[r_n]$ とにわけて

$$\begin{cases} \hat{y}_n & := & \mathrm{midpoint}([y_n]), \\ [r_n] & := & [y_n] - \hat{y}_n \end{cases} \tag{5}$$

とし，それぞれの 1 ステップの積分による移り先を別々に計算する方法である:

$$\begin{cases} \hat{y}_{n+1} & := & \hat{y}_n + h\Phi(\hat{y}_n; h) + \hat{z}_{n+1}, \\ [r_{n+1}] & := & [A_n][r_n] + [z_{n+1}] - \hat{z}_{n+1}. \end{cases} \tag{6}$$

$\hat{z}_{n+1}$ は $[z_{n+1}]$ の中点であり，$[A_n]$ は $[r_n]$ が n 番目のステップでの積分によりどのように変化するかを表す区間 matrizant である．

本稿では matrizant $[A_n]$ の計算法として Moore の方法 [4] と Lohner の方法 [3] を用いる．Moore の方法は，行列微分方程式の初期値問題

$$\frac{\mathrm{d}C}{\mathrm{d}t}(t,\xi) = \frac{\partial f}{\partial y}C(t,\xi), \quad C(t_n,\xi) = I \ (I : \text{unit matrix}, \ \xi : \text{定数}) \tag{7}$$

の解 $C(t,\xi)$ を用いて，matrizant $[A_n]$ を

$$[A_n] := C(t_n + h, [y_n]) \tag{8}$$

で決める方法である．本稿では初期値問題 (7) を 1 次の剰余項まで展開する Taylor 展開法を用いて保証区間付で解き matrizant $[A_n]$ とした．Lohner の方法は，matrizant $[A_n]$ を

$$[A_n] := I + \frac{\partial \Phi}{\partial \xi}([y_n]; h) \tag{9}$$

で決める方法である．

3 計算手順

本稿で構成した次数とステップ幅自動調節機能付 Taylor 展開法は，以下に示すような手順により 1 ステップの積分を行なう．

手順 1　未知関数の変動区間を見積もる．
引数: ステップ幅 h_{next}, 数値解 $\hat{y}_n$, 全離散化誤差の評価区間 $[r_n]$.
出力: ステップ幅 h, 解の変動区間の見積もり Y_n.

手順 2　次数とステップ幅を決定する．
引数: ステップ幅 h, 次数 p, 解の変動区間の見積もり Y_n.
出力: ステップ幅 h, 次数 p, 次のステップでのステップ幅 h_{next}.

手順 3　剰余項の評価区間を計算する.
引数: ステップ幅 h, 次数 p, 解の変動区間の見積もり Y_n.
出力: 剰余項の評価区間 $[z_{n+1}]$.

手順 4　伝播誤差を評価するための matrizant を計算する.
引数: ステップ幅 h, 次数 p, 数値解 $\hat{y}_n$, 全離散化誤差の評価区間 $[r_n]$,
解の変動区間 Y_n.
出力: matrizant $[A_n]$.

手順 5　数値解を保証付きで求める.
引数: ステップ幅 h, 次数 p, 数値解 $\hat{y}_n$, 全離散化誤差の評価区間 $[r_n]$,
剰余項の評価区間 $[z_{n+1}]$, matrizant $[A_n]$.
出力: 数値解 $\hat{y}_{n+1}$, 全離散化誤差の評価区間 $[r_{n+1}]$.

4　未知関数の変動区間を見積もる方法

剰余項を評価するために $[t_n, t_n+h]$ における y の変動区間の見積もり Y_n が必要となる. 見積もりには常微分方程式の解の存在定理である Cauchy-Peano の定理を応用する.

定理 1　[Cauchy-Peano の定理]　f は R^{n+1} の閉領域

$$E := \{(t,y) : |t-t_n| \le h, \| y-y_n \| \le \rho\}$$

で連続とする. f は E において有界である:

$$\| f(t,y) \| \le M.$$

そのとき, 初期条件を満たす解が区間

$$|t-t_n| \le \min(h, \rho/M)$$

において存在する.

[証明]　略す.　■

変動区間を得る手法の原理は次のように考えると分かりやすい. まず $\|\cdot\|$ を各成分に重み w_i をつけたノルムとする:

$$\|x\| = \max_i |w_i x_i|, \quad x \in R^d. \tag{10}$$

Y_{init} は中点が y_0 となるようにとり, 重み w_i の値は, Y_{init} の各成分に対し各成分の区間幅の逆数とする. 次に, 閉領域 $[t_n, t_n+h] \times Y_{\text{init}}$ における f の変動区間 $W(f, Y_{\text{init}})$ は $f(Y_{\text{init}})$ に含まれる:

$$f(Y_{\text{init}}) \supseteq W(f, Y_{\text{init}}).$$

もし

$$Y_{\text{init}} \supseteq y_n + [0,h]f(Y_{\text{init}}) \supseteq y_n + [0,h]W(f,Y_{\text{init}}) \tag{11}$$

なら，平均値の定理より $\min(h,\rho/M) = h$ となり，閉領域 $[t_n,t_n+h]\times Y_{\text{init}}$ における解の存在が保証される．すなわち，解の変動区間は Y_{init} に含まれる．

具体的な手続きを述べる．まず適当な閉区間 Y_{init} を選ぶ．次に $\bar{Y}_{\text{init}} := y_n+[0,h]f(Y_{\text{init}})$ とし，$Y_{\text{init}} \supseteq \bar{Y}_{\text{init}}$ が成り立つかどうかチェックする．成り立てば $[t_n,t_n+h]\times Y_{\text{init}}$ に解が存在し，変動区間の見積もり $Y_n := Y_{\text{init}}$ を得る．成り立たなければ Y_{init} の幅を少し広げ再度 $\bar{Y}_{\text{init}}$ を計算する．Y_{init} の幅を少し広げ $\bar{Y}_{\text{init}}$ を計算するという操作を数回繰り返しても $Y_{\text{init}} \supseteq \bar{Y}_{\text{init}}$ が成り立たなければ，今度はステップ幅 h を 1/2 倍して再び閉区間を選択し $\bar{Y}_{\text{init}}$ の計算から始める．

また変動区間の見積もり Y_n が得られたなら，

$$Y_n := y_n + [0,h]f(Y_n) \tag{12}$$

を何回か繰り返しさらに変動区間の縮小を試みる．

本稿では具体的に次のような方法を用いた．

手続き 1

```
    begin
1       Y_init := [y_n] + [-2h, 2h]f([y_n]);
2       counter := 0; Y_tmp = Y_init;
3       Ȳ_init := [y_n] + [0, h]f(Y_init);
4       while Ȳ_init ⊈ Y_init do
            begin
5               Y_init := (1 + 0.1)Y_init − 0.1Y_init;
6               Ȳ_init := [y_n] + [0, h]f(Y_init);
                if counter < 5 then counter := counter + 1
                else
                    begin
7                       counter := 0; h := h/2; Y_init := Y_tmp;
8                       Ȳ_init := [y_n] + [0, h]f(Y_init)
                    end
            end;
9       Y_n := Ȳ_init
    end
```

---

5 行目は区間の幅を 1.2 倍する．この手続きで出力されるのは未知関数の変動区間 Y_n と 7 行目で変更されたステップ幅 h である．未知関数は $[0,h]\times Y_n$ で解の存在が保証されたことになる．

5 次数とステップ幅の決定法と剰余項の評価区間の計算

各ステップにおける局所打切誤差が一定値 ε 以下になるようにステップ幅 h は決める.

各ステップごとに何らかの方法で次数 $p_{\max}$ を決め，それ以下の全ての次数 q に対して次の関係式を満たす h_q を求めておく:

$$h_q^{q+1} width(y^{<q+1>}(Y_n)) = \varepsilon h_q. \tag{13}$$

$width([\cdot])$ は $[\cdot]$ の成分の区間幅の最大値を表す．本稿では前ステップでの次数 p がある次数 p_{MAX} より小さければ $p_{\max} = p+1$ とし，p_{MAX} と等しければ $p_{\max} = p_{\mathrm{MAX}}$ とした．次に，求めた q, h_q $(q = 1, \ldots, p_{\max})$ の組の中から (i)h_q が最大である組を選ぶ方法; (ii)h_q/q^2 が最大である組を選ぶ方法を用いて適切な組を選び，選んだ q と $\min(h, h_q)$ をそのステップで用いる次数 p とステップ幅 h とする．t を単位時間進める際，方法 (i) ではステップ数が最小となり，方法 (ii) では計算時間が短くなることが期待できる．なぜなら，t を単位時間進める際のステップ数は $\mathrm{O}(1/h)$ であり，1 ステップに要する計算量は p 次まで Taylor 展開すれば $\mathrm{O}(p^2)$ となるので，(ii) のように h_q/q^2 が最大になるような組を選べば t を単位時間進める際の計算時間は (i) に比べれば短くなると期待できるからである.

本稿では具体的に次のような方法を用いた.

手続き 2

begin

1 **for** $q := 1$ **until** $p_{\max}$ **do** $h_q := \left(\dfrac{\varepsilon}{width(y^{<q+1>}(Y_n))} \right)^{\frac{1}{q}}$;

2 (i) $p := \arg\max_q \{h_q \quad : q = 1, 2, \ldots, p_{\max}\}$;

(ii) $p := \arg\max_q \{h_q/q^2 : q = 1, 2, \ldots, p_{\max}\}$;

3 $h_{\mathrm{next}} := h_p$;

4 $h := \min\{h, h_{\mathrm{next}}\}$

end

---

h をそのステップでのステップ幅とし h_{next} を次のステップでのステップ幅の予測とする.

Y_n を引数とした未知関数 y の $p+1$ 次の Taylor 展開係数 $y^{<p+1>}(Y_n)$ は既に計算してあるので剰余項の評価区間 $[z_{n+1}]$ は次のようになる:

$$[z_{n+1}] := h^{p+1} y^{<p+1>}(Y_n). \tag{14}$$

6 Matrizant の計算法

Moore の方法における matrizant の計算法を示す.

ξ を固定すると，$C(t, \xi)$ は次の行列微分方程式の初期値問題の解である:

$$\frac{\mathrm{d}C}{\mathrm{d}t}(t, \xi) = \frac{\partial f}{\partial y} C(t, \xi), \quad C(t_n, \xi) = I. \tag{15}$$

ここで必要なものは $C(t_n+h,\xi)$ なので，初期値問題 (15) を区間 $[t_n,t_n+h]$ において，精度保証付きで数値的に解き matrizant $[A_n]$ とすればよい．

本稿において (15) を解くのに用いた数値解法は，1 次の剰余項まで展開する Taylor 展開法である．すなわち

$$\begin{aligned} C(t_n+h,\xi) &= I+h\left[\frac{\partial f}{\partial y}C(t,\xi)\right]_{t=t_n+\theta h}, \\ &\in I+h\left[\frac{\partial f}{\partial y}C(t,\xi)\right]_{t=[t_n,t_n+h]}, \\ &= I+h\frac{\partial f}{\partial y}(y([t_n,t_n+h],\xi))C([t_n,t_n+h],\xi) \end{aligned}$$

より，前章と同様にして C の変動区間 $[C_{\text{init}}]$ を見積もり次を得る:

$$C(t_n+h,[y_n]) \subseteq I+h\frac{\partial f}{\partial y}(y([t_n,t_n+h],t_n,[y_n]))[C_{\text{init}}]. \tag{16}$$

$y([t_n,t_n+h],t_n,[y_n])$ については，既に求めた y の変動区間 Y_n を用いた．

本稿では具体的に次のような方法を用いた．

手続き 3

begin

1 $[C_{\text{init}}] := I+[-2h,2h]\dfrac{\partial f}{\partial y}(Y_n)$; $counter=0$;

2 $[\bar{C}_{\text{init}}] := I+[0,h]\dfrac{\partial f}{\partial y}(Y_n)[C_{\text{init}}]$;

3 **while** $[\bar{C}_{\text{init}}] \not\subseteq [C_{\text{init}}]$ **do**

begin

4 $[C_{\text{init}}] := (1+0.1)[C_{\text{init}}]-0.1[C_{\text{init}}]$;

5 $[\bar{C}_{\text{init}}] := I+[0,h]\dfrac{\partial f}{\partial y}(Y_n)[C_{\text{init}}]$;

6 **if** $counter<20$ **then** $counter := counter+1$

7 **else** program をぬける

end;

8 $[A_n] := I+h\dfrac{\partial f}{\partial y}(Y_n)[C_{\text{init}}]$

end

上に示した手法は 3 行目から 7 行目までの while ループが不完全であるが，以後に行なった実験ではここでプログラムが止まることはなかった．

行列微分方程式の変動区間を求めるときの積分区間を $[t_n,t_n+h/m]$,$[t_n+h/m,t_n+2h/m]$, ...,$[t_n+(m-1)h/m,t_n+h]$ と細分し，それぞれの区間で行列微分方程式を解いて，各区間から得られた matrizant の積をつくり第 n ステップの matrizant $[A_n]$ とする方法もある．

Lohner の方法では matrizant $[A_n]$ を次のように与える:

$$[A_n] := I + h\frac{\partial \Phi}{\partial \xi}([y_n]; h) \tag{17}$$

自動微分法を用いれば Φ を計算する手続きから直接 $\partial\Phi([y_n]; h)/\partial\xi$ を求めることができるが，本稿では別の方法 [3] を用いた．その方法は，Φ が p 次の Taylor 展開なら行列微分方程式の初期値問題

$$\frac{\mathrm{d}C}{\mathrm{d}t}(t, [y_n]) = \frac{\partial f}{\partial y}C(t, [y_n]), \quad C(t_n, [y_n]) = I \tag{18}$$

の解 C の，$t = t_n$ における p 次までの Taylor 展開が右辺となることを用いる．

本稿では C の $t = t_n$ における Taylor 展開の係数を次のように計算した．まず，y の $t = t_n$ における Taylor 展開係数を，C の p 次の展開係数が必要ならば p 次まで自動微分法を用いて計算しておく．計算する際の引数は $[y_n]$ である．関数 $\partial f/\partial y$ の計算手続きを与えておけば，y の p 次までの展開係数と C の初期値 I より，初期値問題 (18) の解 C の Taylor 展開係数が自動微分法を用いて計算できる．

この方法は $[y_n]$ を引数として C の Taylor 展開係数を計算しているので，$t = t_n$ における Taylor 展開係数 $C^{<p>}$ を，引数を $[y_n]$ として計算したことを示すために $C^{<p>}([y_n])$ と記す．

以上を式にすると

$$I + h\frac{\partial \Phi}{\partial \xi}([y_n]; h) = I + \sum_{j=1}^{p} h^j C^{<j>}([y_n]). \tag{19}$$

となり，次の手続きを得る．

手続き 4

begin

1 $[y_n]$ を引数として，y の p 次までの Taylor 展開係数を計算する;

2 y の Taylor 展開係数を用いて，C の p 次までの Taylor 展開係数を計算する;

3 $[A_n] := I + \sum_{j=1}^{p} h^j C^{<j>}([y_n])$

end

---

7 精度保証付き数値解の計算

n 番目のステップでの局所打切誤差の評価区間 $[z_{n+1}]$ と matrizant $[A_n]$,n 番目のステップ点での数値解 $\hat{y}_n$ と全離散化誤差の評価区間 $[r_n]$ を用いて，$n+1$ 番目のステップ点での数値解 $\hat{y}_{n+1}$ と全離散化誤差の評価区間 $[r_{n+1}]$ を計算する:

$$\begin{cases} \hat{y}_{n+1} & := \hat{y}_n + h\Phi(\hat{y}_n; h) + \hat{z}_{n+1}, \\ [r_{n+1}] & := [A_n][r_n] + [z_{n+1}] - \hat{z}_{n+1}. \end{cases} \tag{20}$$

8 wrapping effect

式 (20) において全離散化誤差の評価区間を計算する際，各ステップごとに

$$[r_{n+1}] := [A_n][r_n] + [z_{n+1}] - \hat{z}_{n+1} \tag{21}$$

としているが，この漸化式を用いると保証区間が過度に拡大することがある．これは次のような理由による．d 次元直方体領域 $[r_n]$ は区間行列 $[A_n]$ により一次変換すると d 次元平行多面体領域に移る．しかし，$[A_n][r_n]$ はこの d 次元平行多面体領域を含む区間ベクトル，すなわち d 次元直方体領域になり，$[A_n][r_n]$ は本来より大きな領域になる．このように漸化式 (21) を用いて保証区間の計算を行なうと，ステップごとに余計な領域が保証区間に加えられて保証区間が過度に拡大するのである．この現象を wrapping effect とよぶ [3].

wrapping effect による保証区間の過度の拡大は $[r_{n+1}]$ の計算法を次のようにすれば抑えることができよう:

$$\begin{cases} [r_{n+1}] & := & \displaystyle\sum_{i=0}^{n+1} [A_{n+1,i}]([z_i] - \hat{z}_i), \\ [A_{n+1,i}] & := & [A_n][A_{n,i}], \quad i = 0, \ldots, n, \ [A_{n,n}] = I. \end{cases} \tag{22}$$

(22) では第 n ステップの積分を行なうのに区間ベクトル $[z_i] - \hat{z}_i$ と区間行列 $[A_{n+1,i}]$ とのかけ算が 1 回だけで済む．よって $[A_n]$ の各成分の区間幅が 0 ならば wrapping effect はおこらない．しかし (22) では第 n ステップの積分を行なうのに区間行列同士のかけ算を n 回，区間行列と区間ベクトルのかけ算を n 回行わなければならず，計算にかかる時間及び計算に必要な記憶領域はステップ数とともに増大する．

9 数値実験 (Swingby)

例として太陽，木星，地球の引力の影響下で運動する宇宙船の運動方程式

$$\begin{cases} \dfrac{\mathrm{d}x}{\mathrm{d}t} &= u, \\ \dfrac{\mathrm{d}y}{\mathrm{d}t} &= v, \\ \dfrac{\mathrm{d}u}{\mathrm{d}t} &= -\dfrac{x}{\{x^2+y^2\}^{\frac{3}{2}}} - Gm\dfrac{(x - r\cos\omega t)}{\{(x - r\cos\omega t)^2 + (y - r\sin\omega t)^2\}^{\frac{3}{2}}} \\ & \quad -GM\dfrac{(x - \cos(t+\phi))}{\{(x - \cos(t+\phi))^2 + (y - \sin(t+\phi))^2\}^{\frac{3}{2}}}, \\ \dfrac{\mathrm{d}v}{\mathrm{d}t} &= -\dfrac{y}{\{x^2+y^2\}^{\frac{3}{2}}} - Gm\dfrac{(y - r\sin\omega t)}{\{(x - r\cos\omega t)^2 + (y - r\sin\omega t)^2\}^{\frac{3}{2}}} \\ & \quad -GM\dfrac{(y - \sin(t+\phi))}{\{(x - \cos(t+\phi))^2 + (y - \sin(t+\phi))^2\}^{\frac{3}{2}}} \end{cases} \tag{23}$$

を初期条件

$$x(0) = 0.19004, \ y(0) = 0.0, \ u(0) = 1.95, \ v(0) = 2.28 \tag{24}$$

のもとで解く問題を取り上げた．ここで，座標系 (x,y) は太陽を原点とする慣性座標系をとり，地球と木星は太陽の周りを同一平面内で円運動をすると仮定し，そして，太陽・木星間距離が 1 となるように長さの単位を，木星の角速度が 1 となるように時間の単位を，太陽の質量が 1 となるように質量の単位を，それぞれ選んだ．Gを万有引力定数，mと Mをそれぞれ地球と木星の質量とし，$Gm=3.0404\times10^{-6}$, $GM=9.5479\times10^{-4}$, $r=0.19$, $\omega=12.0$ と置いた．また，地球と木星との相互位置関係は $\phi=0.4835$ で設定した．

10 解法の比較

この初期値問題を保証区間付きで数値的に解いた．使用計算機は SUN Microsystems 社の SPARC Station，言語は GNU C++ である．浮動小数点データ形式は IEEE 倍精度形式を用いた．次数とステップ幅を決める方法については，前章で示したふたつの方法: (i) ステップ幅 h 最大; (ii) h/p^2 最大の比較を行った．局所打切誤差の上界は $\varepsilon=10^{-10}$ に設定した．また，保証区間を計算する方法として次の方法について比較を行った:

(a) 保証区間をもっとも簡単な方法で計算する:

$$[y_{n+1}] := [y_n] + h\Phi([y_n];h) + [z_{n+1}]; \tag{25}$$

(b) 保証区間を Moore の方法で求めた matrizant $[A_n]$ を用いて計算する:

$$\begin{cases} \tilde{y}_{n+1} &:= \tilde{y}_n + h\Phi(\tilde{y}_n;h) + \tilde{z}_{n+1}, \\ [r_{n+1}] &:= [A_n][r_n] + [z_{n+1}] - \tilde{z}_{n+1}; \end{cases} \tag{26}$$

(b′) 保証区間を matrizant を用いて計算する際に，matrizant $[A_n]$ は Moore の方法で定めるが，行列微分方程式を解く区間を $[t_n, t_n+h/2]$, $[t_n+h/2, t_n+h]$ に 2 分割し，それぞれの区間で得られた保証区間付き数値解の積を $[A_n]$ とする;

(c) 保証区間を matrizant を用い公式 (26) により計算するが，matrizant $[A_n]$ は Lohner の方法で求めた．

(b′1) 保証区間を matrizant を用いて計算する際に，matrizant $[A_n]$ は (b′) と同じ方法を用いて計算するが，$[r_{n+1}]$ は次の方法で計算する:

$$\begin{cases} [r_{n+1}] &:= \displaystyle\sum_{i=0}^{n+1} [A_{n+1,i}]([z_i] - \tilde{z}_i), \\ [A_{n+1,i}] &:= [A_n][A_{n,i}], \quad i=0,\ldots,n,\ [A_{n,n}] := I. \end{cases} \tag{27}$$

(c1) 保証区間を matrizant を用いて計算する際に，matrizant $[A_n]$ は Lohner の方法と同様に計算するが，$[r_{n+1}]$ は (27) で計算する:

11 実験結果

図 1は地球の軌道，木星の軌道，宇宙船の軌道を北極方向からみた図と，木星の軌道を横切る部分の拡大図である．

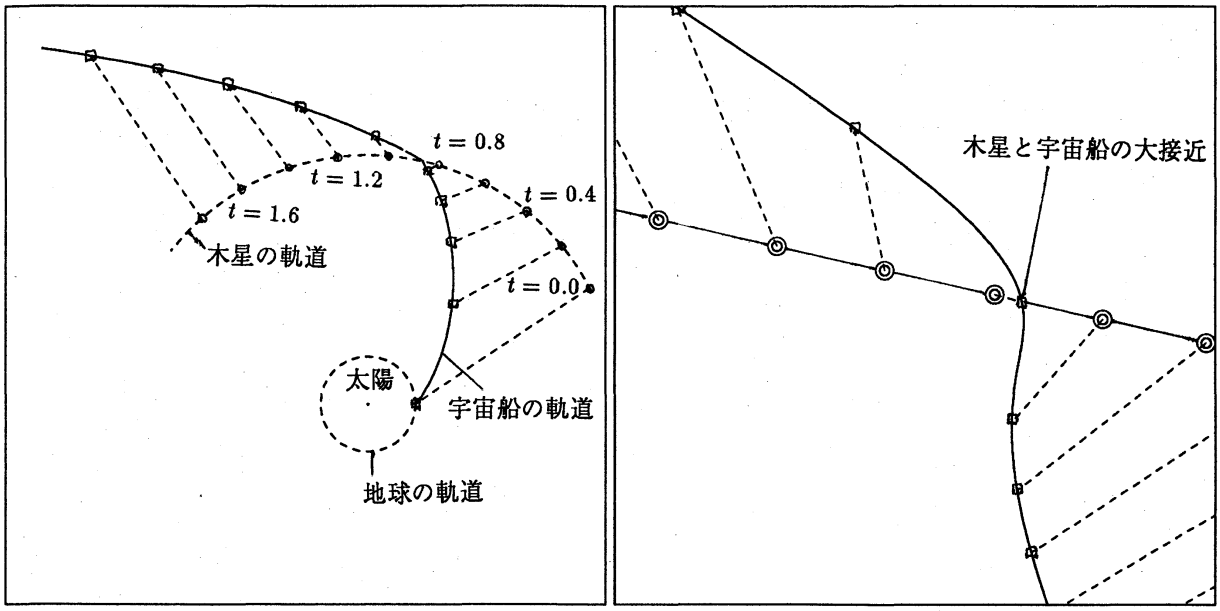

図 1. 軌道図

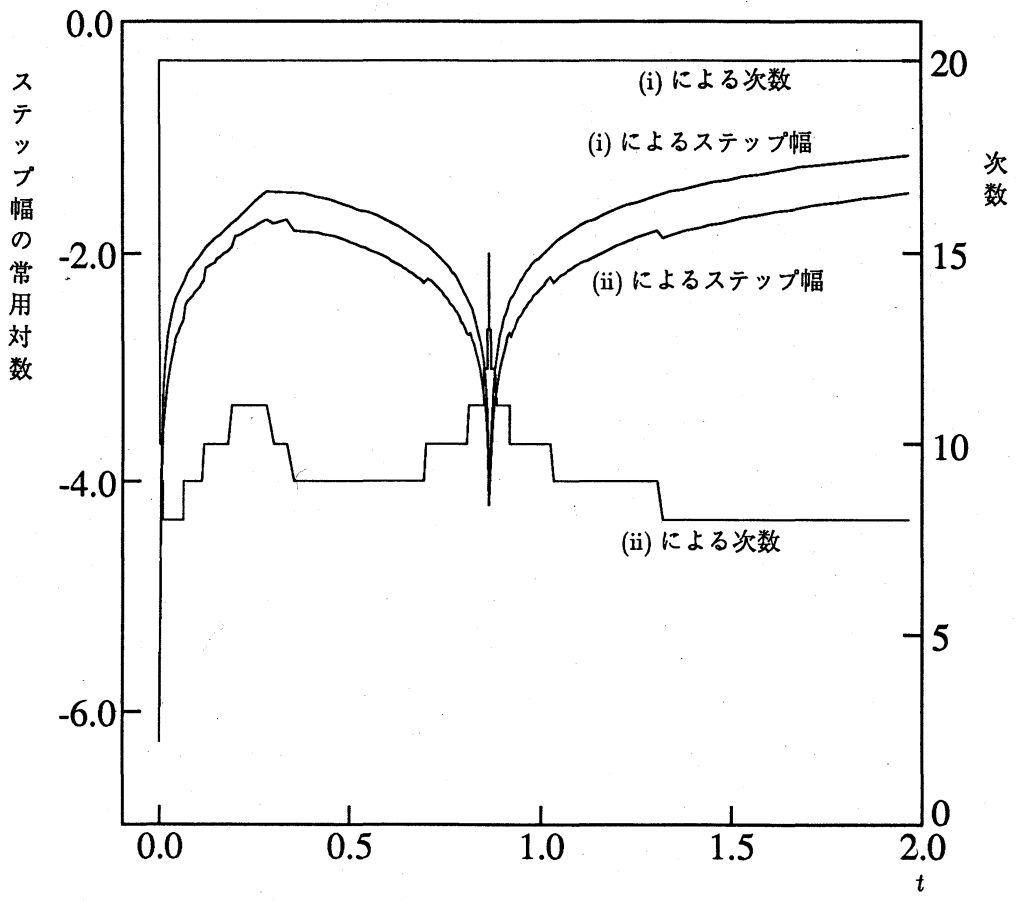

図 2. 次数及びステップ幅の変化

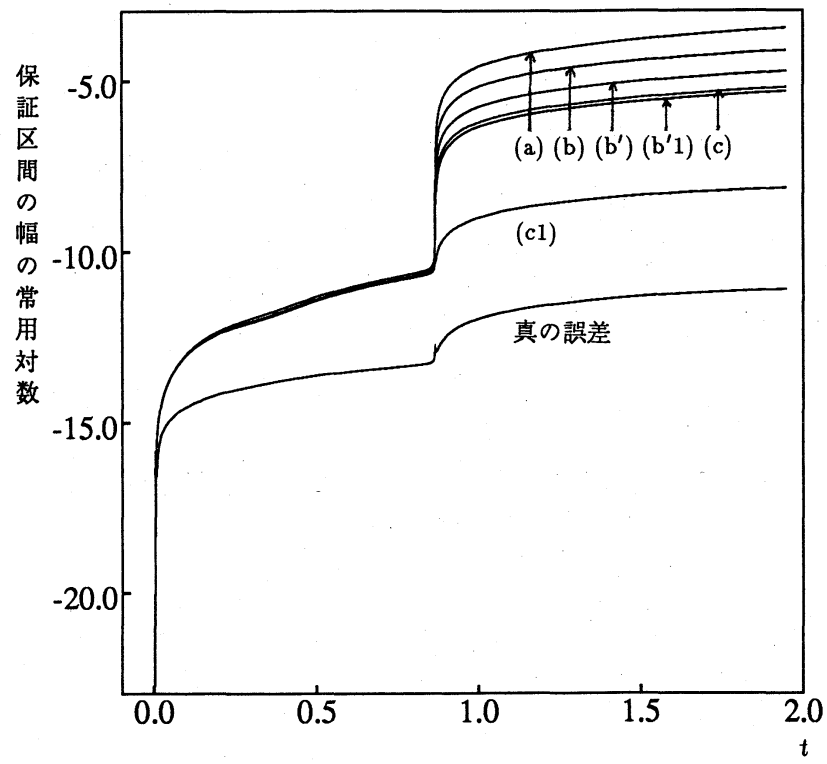

図 3. (i) による保証区間の幅の変化

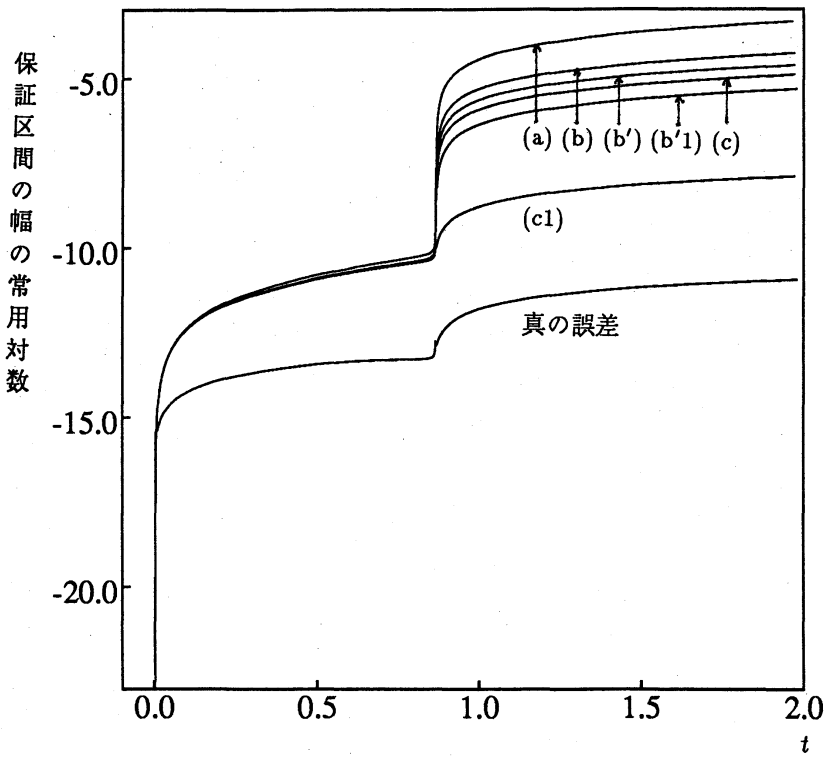

図 4. (ii) による保証区間の幅の変化

(i), (ii) の方法を用いて次数とステップ幅が変化する様子を図2に示す．変化の様子を示すグラフが図2だけである理由は，保証区間の計算法の違いによる次数とステップ幅の変化の違いがこの積分区間 $[0,2]$ ではほとんどなかったからである．

それぞれの保証区間の計算法を用いて保証区間の幅が変化する様子を図3と図4に示す．図3, 4は次数とステップ幅をそれぞれ (i), (ii) の方法を用いて決定したときの変化の様子である．このグラフの中に示してある“真の誤差”とは，初期値問題を (c) の方法を用いて解いたときの数値解の中心点と，全てのステップにおけるステップ幅を 1/2 にして倍のステップ数をかけて解いたときの数値解の中心点との差の絶対値のことである．

各方法を用いたときの計算に要した時間を表1に示す．

表 1. 計算に要した時間 (秒)

表の外の中の上にある数字はは CPU 時間であり，下の $(\cdot,\cdot)$ は CPU 時間の比である．(α,β), α: p,h の計算に同じ方法を用いた時，保証区間の計算に (a) を用いた時を 1 とした，他の計算法との所要時間の比，β: 保証区間の計算に同じ方法をもちいた時，p,h の計算に (i) を用いた時を 1 とした，他の計算法との所要時間の比．

p と h の計算法	保証区間の計算法					
	(a)	(b)	(b′)	(b′1)	(c)	(c1)
(i)	309.9	372.5	419.3	881.3	963.8	1452.6
372steps	(1,1)	(1.2,1)	(1.4,1)	(2.8,1)	(3.1,1)	(4.7,1)
(ii)	229.7	294.1	351.2	1977.2	664.7	2293.1
680steps	(1,0.7)	(1.3,0.8)	(1.5,0.8)	(8.6,2.2)	(2.9,0.7)	(10,1.6)

12 Runge-Kutta 法との比較

比較のため4段4次 Runge-Kutta 法にステップ幅調節機能をつけ同じ初期値問題を解いた．ステップ幅の自動調節は次のように行なう．まずステップ幅 h で1ステップ進んだ時の数値解を $y1$ とし，ステップ幅 $h/2$ で2ステップ進んだ時の数値解を $y2$ とする．すると $error = |y1 - y2|/h$ を局所打切誤差の推定値として使える．$error$ がある値 $\varepsilon_{\max}$ より大きければステップ幅を 1/2 倍して数値解を計算し直す．また $error$ がある値 $\varepsilon_{\min}$ より小さければ現在のステップ幅で1ステップ進み，次のステップでのステップ幅を2倍にする．

表2に数値計算の結果を示す．

13 考察

図3と図4を見ると (c1) 以外の方法による保証区間の拡大が非常にはげしいことがわかる．この様子は wrapping effect によるものと解釈できよう．(b′) に wrapping effect 対策を施した (b′1) が (c) に対する (c1) ほど効果がないのは，(b′) の区間 matrizant の各成分の

表 2. 数値計算結果

積分区間を $[0,2]$ とし $\varepsilon_{\max}$ の値を変えてそれぞれステップ数，所要時間，位置座標を調べた．有効数字とは，数字のある行の位置座標の数値解をその下の行の位置座標の数値解と比べたときの，一致する有効数字の個数である．

$\varepsilon_{\max}$	steps	CPU time (秒)	有効数字	位置	
				x	y
10^{-5}	729	2.7	4	−1.3032	1.4292
10^{-6}	1293	4.6	5	−1.30341	1.42907
10^{-7}	2225	7.9	6	−1.303436	1.429056
10^{-8}	3733	13.3	-	−1.303438	1.429054

区間幅が (c) の区間 matrizant のよりも大きくて，matrizant の積を計算すると積の成分の区間幅が激しく拡大してしまうからと推測できる．

次に，表 2の有効数字を全離散化誤差の目安として，Taylor 展開法と計算時間についての比較をする．まず Runge-Kutta 法を用いたとき有効数字 4 桁を実現するのは，すなわち全離散化誤差が 10^{-3} で抑えられるのは $\varepsilon_{\max} = 10^{-5}$ を選んだ場合であり，数値解の計算と誤差の推測に (2.7+4.6) 秒かかった．これに見合う精度保証ができる Taylor 展開法は (i,a) と (ii,a) の組み合わせである．この中で最も計算時間が短いのは (ii,a) の組み合わせであり，計算時間は Runge-Kutta 法の 40 倍程度である．同様の比較を $\varepsilon_{\max} = 10^{-6}$ (有効数字 5 桁) に対応する (ii,b) で行なえば，Taylor 展開法は Runge-Kutta 法の 36 倍程度の時間で数値解と正しい誤差の上限が得られることがわかる．

14 おわりに

自動微分法を用いて常微分方程式の初期値問題を精度保証付で解く方法についていくつかの手法を計算機上に構成して数値実験をした．また従来の方法との比較も行なった．その結果ここで用いたステップ幅調節機能付 Runge-Kutta 法のおおよそ数十倍の計算時間をかけて，この Runge-Kutta 法で得られる全離散化誤差の“目安”と同程度の，“厳密な誤差の上限”が与えられた数値解を Taylor 展開法により得ることができた．また，本稿の方法 (c1) は計算時間と計算領域は大量に必要だが，本稿で試した他の方法に比べて極めて狭い保証区間を与えることができるので，非常に条件の悪い問題を解く場合には有効であろう．

参考文献

[1] 伊理正夫, 久保田光一: 高速自動微分法 (I);(II). 応用数理, Vol. 1, No. 1 (1991 年 3 月), pp. 17–35; No. 2 (1991 年 6 月), pp. 53–63.

[2] L. B. Rall: *Automatic Differentiation—Techniques and Applications*. Lecture Notes in Computer Science 120, Springer-Verlag, 1981.

[3] R. J. Lohner: Enclosing the Solutions of Ordinary Initial and Boundary Value Problems. In E. Kaucher, U. Kulisch, and Ch. Ullrich (eds.): *Computer Arithmetic, Scientific Computation and Programming Languages*, B. G. Teubner, Stuttgart, 1987, pp. 255–286.

[4] R. E. Moore: Automatic Local Coordinate Transformations to Reduce the Growth of Error Bounds in Interval Computation of Solutions of Ordinary differential Equations. In L. B. Rall (ed.): *Error in Digital Computation*, Vol. 2, John Wiley & Sons, Inc., New York, 1965, pp. 103–140.

[5] R. E. Moore: *Methods and Applications of Interval Analysis*. SIAM, Philadelphia, 1979.

[6] H. J. Stetter: Validated Solution of Initial Value Problems for ODE. In Ch. Ullrich (ed.): *Computer Arithmetic and Self-Validating Numerical Methods*, Notes and Reports in Mathematics in Science and Engineering, Vol. 7, Academic Press, 1990, pp. 171–187.

[7] 雨宮治郎, 谷口行信: 自動微分法を取り入れた常微分方程式の諸数値解法の比較: 解法の効率と精度の保証. 第 20 回数値解析シンポジウム講演予稿集, pp. 182–185, 1991.

[8] 雨宮治郎: Taylor 展開に基づく常微分方程式の数値解法. 日本応用数理学会平成 3 年度年会研究発表予稿集, pp. 106–107, 1991.